Claim Rejections - 35 USC §102

Claims 1, 4 – 5, 15, 17, 20-21, 31-33, 36-37 and 47 were rejected under 35 U.S.C. 102(e) as being anticipated by Fradette (US PAT 6,606,698). This rejection is respectfully traversed.

The Applicant's exemplary Claim 1 sets forth:

"A memory interface device for interfacing a number of host applications to a memory device, the memory interface device comprising:

a host interface for interfacing with the number of host applications in a protocol associated with the corresponding host application;

a memory interface for interfacing with the memory device wherein one or more of the host applications and the memory device operate in response to different protocols;

a number of contexts operably coupled to the host interface for receiving memory access requests from the number of host applications and providing result/status information to the number of host applications, wherein at least one context is provided for each host application; and

control logic operably coupled to obtain memory access requests from the number of contexts, translate the memory access requests into memory access requests in accordance with a protocol of the memory device, interact with the memory device over the memory interface for servicing the memory access requests on behalf of the number of host applications, and provide the result/status information to the number of host applications via the number of contexts in accordance with the protocol associated with each of the number of host applications."

An implementation of the Applicants' invention operates to translate requests between multiple different host applications and a memory. Each host application has at least one of its own contexts that stores the requests and result status information. The multiple request and status contexts can be advantageous for flexible memory accesses.

In contrast, Fradette discloses a data storage managing apparatus for moving data between clients and data storage. The data storage managing apparatus includes a request processing apparatus 60. The request processing apparatus 60 includes multiple interface modules 120, one for each host storage protocol (Col. 3 lines 54 – 63). The request processing apparatus 60, however, contains only a single request queue 122 for storing requests for each host interface

module 120 (Col. 4 lines 13 – 14). Thus Fradette fails to teach the Applicants' claimed multiple contexts "operably coupled to the host interface for receiving memory access requests from the number of host applications and providing result/status information to the number of host applications...".

The Office Action suggests that the Applicants' "contexts" are represented by memory map 124 in request processing apparatus 60. The Applicants disagree. The Applicants have clearly claimed that each context is "operably coupled to the host interface for receiving memory access requests from the number of host applications and providing result/status information to the number of host applications...". In contrast, the memory map 124 of Fradette is used only for storage of data (Col. 4 lines 18 – 19, Col. 4 lines 43 – 45). For these reasons, Fradette fails to teach or suggest the Applicant's claimed invention including multiple contexts for receiving memory access requests and providing result/status information. The Applicant therefore respectfully asserts that Claim 1, and its dependent claims 4 – 5 and 15 are in condition for allowance.

The Applicant's independent claims 17 and 32 contain limitations analogous to those of claim 1. The Applicant therefore respectfully asserts that claim 17 and its dependent claims 20-21 and 31, and claim 32 and its dependent claims 33, 36-37 and 47 are also in condition for allowance.

Claim Rejections – 35 USC §103

Claims 2, 16, 18, 34, and 48 were rejected under 35 U.S.C. 103(a) as being unpatentable over Fradette in view of Wentka et al. (US PAT 5,968,114). This rejection is respectfully traversed.

Claims 2 and 16 are dependent upon claim 1. Claim 18 is dependent upon claim 17. Claims 34 and 48 are dependent upon claim 32. As previously set forth, Fradette fails to teach or suggest the multiple contexts for receiving memory access requests and providing result/status information. Wentka adds nothing further to Fradette that would suggest the claimed multiple contexts. Since Fradette and Wentka, taken alone or in combination, fail to teach or suggest the claimed contexts, the Applicant respectfully asserts that claims 2, 16, 18, 34, and 48 are allowable for the reasons set forth with regard to claim 1.

Claims 3, 19, and 35 were rejected under 35 U.S.C. 103(a) as being unpatentable over Fradette in view of Bauman et al. (US PAT 5,875,472). This rejection is respectfully traversed.

Claim 3 is dependent upon claim 1. Claim 19 is dependent upon claim 17. Claim 35 is dependent upon claim 32. As previously set forth, Fradette fails to teach or suggest the multiple contexts for receiving memory access requests and providing result/status information. Bauman adds nothing further to Fradette that would suggest the claimed multiple contexts. Since Fradette and Bauman, taken alone or in combination, fail to teach or suggest the claimed contexts, the Applicant respectfully asserts that claims 3, 19, and 35 are allowable for the reasons set forth with regard to claim 1.

Claims 6 – 7, 9 – 10, 12, 22 – 23, 25 – 26, 28, 38 – 39, 41 – 42, and 44 were rejected under 35 U.S.C. 103(a) as being unpatentable over Fradette in view of Hughes et al. (US PAT 5,584,582). This rejection is respectfully traversed.

Claims 6 – 7, 9 – 10, and 12 are dependent upon claim 1. Claims 22 – 23, 25 – 26, and 28 are dependent upon claim 17. Claims 38 – 39, 41 – 42, and 44 are dependent upon claim 32. As previously set forth, Fradette fails to teach or suggest the multiple contexts for receiving memory access requests and providing result/status information. Hughes adds nothing further to

Fradette that would suggest the claimed multiple contexts. Since Fradette and Bauman, taken alone or in combination, fail to teach or suggest the claimed contexts, the Applicant respectfully asserts that claims 6 – 7, 9 – 10, 12, 22 – 23, 25 – 26, 28, 38 – 39, 41 – 42, and 44 are allowable for the reasons set forth with regard to claim 1.

Claims 8, 11, 13 – 14, 24, 27, 29 – 30, 40, 43, and 45 – 46 were rejected under 35 U.S.C. 103(a) as being unpatentable over Fradette and Hughes, further in view of Bauman. This rejection is respectfully traversed.

Claims 8, 11, and 13 – 14 are dependent upon claim 1. Claims 24, 27, and 29 – 30 are dependent upon claim 17. Claims 40, 43, and 45 – 46 are dependent upon claim 32. As previously set forth, Fradette fails to teach or suggest the multiple contexts for receiving memory access requests and providing result/status information. Hughes and Bauman adds nothing further to Fradette that would suggest the claimed multiple contexts. Since Fradette, Hughes, and Bauman, taken alone or in combination, fail to teach or suggest the claimed contexts, the Applicant respectfully asserts that claims 8, 11, 13 – 14, 24, 27, 29 – 30, 40, 43, and 45 – 46 are allowable for the reasons set forth with regard to claim 1.

Applicants have made a diligent effort to place the claims in condition for allowance. However, should there remain unresolved issues that require adverse action, it is respectfully requested that the Examiner telephone Applicants' Attorney at 978-264-6664 so that such issues may be resolved as expeditiously as possible.

For these reasons, and in view of the above amendments, this application is now considered to be in condition for allowance and such action is earnestly solicited.

Respectfully Submitted,

7/30/04
Date

Mary Steubing, Reg. No. 37,946
Attorney/Agent for Applicant(s)
Steubing McGuinness & Manaras LLP
125 Nagog Park Drive
Acton, MA 01720
(978) 264-6664

Docket No. 2494/103 120-034
Dd: 6/8/04

CLAIMS

1. (currently amended) A memory interfereace device for interfacing a number of host applications to a memory device, the memory interface device comprising:

a host interface for interfacing with the number of host applications in a protocol associated with the corresponding host application;

a memory interface for interfacing with the memory device wherein one or more of the host applications and the memory device operate in response to different protocols;

a number of contexts operably coupled to the host interface for receiving memory access requests from the number of host applications and providing result/status information to the number of host applications, wherein at least one context is provided for each host application; and

control logic operably coupled to obtain memory access requests from the number of contexts ~~in a protocol associated with the corresponding host interface~~, translate the memory access requests into memory access requests in accordance with a protocol of the memory device, interact with the memory device over the memory interface for servicing the memory access requests on behalf of the number of host applications, and provide the result/status information to the number of host applications via the number of contexts in accordance with the protocol associated with each of the number of host applications.

2. (original) The memory interface device of claim 1, wherein the number of host applications comprises a number of packet processing contexts of a packet processor, and wherein the host interface conforms to a packet processor interface.

3. (original) The memory interface device of claim 1, wherein the memory device comprises a content-addressable memory (CAM), and wherein the memory interface conforms to a CAM interface.

4. (original) The memory interface device of claim 1, wherein the number of contexts comprise a number of context registers sets.

5. (original) The memory interface device of claim 4, wherein each context register set corresponds to one and only one of the number of host applications.

6. (original) The memory interface device of claim 1, wherein the control logic comprises:

monitoring logic;

scheduling logic;

memory interface logic; and

result/status logic, wherein:

the monitoring logic is operably coupled to monitor the number of contexts for detecting memory access requests and providing the memory access requests to the scheduling logic;

the scheduling logic is operably coupled to schedule memory access operations for the memory access requests;

the memory interface logic is operably coupled to generate memory interface signals for interfacing with the memory device over the memory interface; and

the result/status logic is operably coupled to provide result/status information to the number of host application(s).

7. (original) The memory interface device of claim 6, wherein each context comprises a context register set, and wherein the monitoring logic is operably coupled to monitor a predetermined register in each context register set to detect a memory access request.

8. (original) The memory interface device of claim 7, wherein the predetermined register comprises an instruction register.

9. (original) The memory interface device of claim 6, wherein the memory interface supports pipelining of memory access operations, and wherein the scheduling logic is operably coupled to pipeline a plurality of memory access requests over the memory interface.

10. (original) The memory interface device of claim 9, wherein the scheduling logic is operably coupled to determine that a plurality of memory access request conflict and execute at least one of the conflicting memory access requests as an atomic operation.

11. (original) The memory interface device of claim 10, wherein the scheduling logic is operably coupled to clear the pipeline in order to execute the conflicting memory access request as an atomic operation.

12. (original) The memory interface device of claim 6, wherein the result/status logic is operably coupled to correlate result/status information information with its corresponding memory access request.

13. (original) The memory interface device of claim 6, wherein the result/status logic is operably coupled to store the result/status information for each memory access request in a corresponding context.

14. (original) The memory interface device of claim 13, wherein each context comprises a validity indicator, and wherein the result/status logic is operably coupled to set the validity indicator in each context when the corresponding memory access is complete and the result/status information is available.

15. (original) The memory interface device of claim 1 embodied as programmed programmable logic device.

16. (original) The memory interface device of claim 1 embodied as an application specific integrated circuit.

17. (currently amended) Program logic for programming a programmable logic device, the program logic comprising:

host interface logic for interfacing with a number of host applications, the host interface logic operating according to a first interface protocol;

memory interface logic for interfacing with a memory device, the memory interface logic operating according to a second, different protocol;

a number of contexts operably coupled to the host interface logic for receiving memory access requests from the number of host applications and providing result/status information to the number of host applications, wherein at least one context is provided for each host application; and

control logic operably coupled to obtain memory access requests from the number of contexts in the first interface protocol, translate the memory access requests into memory access requests in the second interface protocol, interact with the memory device using the memory interface logic for servicing the memory access requests on behalf of the number of host applications, and provide the result/status information to the number of host applications via the number of contexts in accordance with the first interface protocol.

18. (original) The program logic of claim 17, wherein the number of host applications comprises a number of packet processing contexts of a packet processor, and wherein the host interface logic conforms to a packet processor interface.

19. (original) The program logic of claim 17, wherein the memory device comprises a content-addressable memory (CAM), and wherein the memory interface logic conforms to a CAM interface.

20. (original) The program logic of claim 17, wherein the number of contexts comprises a number of context registers sets.

21. (original) The program logic of claim 20, wherein each context register set corresponds to one and only one of the number of host applications.

22. (original) The program logic of claim 17, wherein the control logic comprises:

monitoring logic;

scheduling logic;

memory interface logic; and

result/status logic, wherein:

the monitoring logic is operably coupled to monitor the number of contexts for detecting memory access requests and providing the memory access requests to the scheduling logic;

the scheduling logic is operably coupled to schedule memory access operations for the memory access requests;

the memory interface logic is operably coupled to generate memory interface signals for interfacing with the memory device using the memory interface logic; and

the result/status logic is operably coupled to provide result/status information to the number of host application(s).

23. (original) The program logic of claim 22, wherein each context comprises a context register set, and wherein the monitoring logic is operably coupled to monitor a predetermined register in each context register set to detect a memory access request.

24. (original) The program logic of claim 23, wherein the predetermined register comprises an instruction register.

25. (original) The program logic of claim 22, wherein the memory interface supports pipelining of memory access operations, and wherein the scheduling logic is operably coupled to pipeline a plurality of memory access requests over the memory interface.

26. (original) The program logic of claim 25, wherein the scheduling logic is operably coupled to determine that a plurality of memory access requests conflict and execute at least one of the conflicting memory access requests as an atomic operation.

27. (original) The program logic of claim 26, wherein the scheduling logic is operably coupled to clear the pipeline in order to execute the conflicting memory access request as an atomic operation.

28. (original) The program logic of claim 22, wherein the result/status logic is operably coupled to correlate result/status information with its corresponding memory access request.

29. (original) The program logic of claim 22, wherein the result/status logic is operably coupled to store the result/status information for each memory access request in a corresponding context.

30. (original) The program logic of claim 29, wherein each context comprises a validity indicator, and wherein the result/status logic is operably coupled to set the validity indicator in each context when the corresponding memory access is complete and the result/status information is available.

31. (original) The program logic of claim 17 embodied in a computer readable medium.

32. (previously amended) An apparatus comprising:

a number of host applications;

a memory device, wherein one or more of the host applications and the memory device ~~have~~ operate using different protocols; and

a memory interface device interposed between the host applications and the memory device and operably coupled to receive memory access requests from the number of host applications, translate the memory access requests into requests in accordance with a protocol of the memory device, interact with the memory device on behalf of the number of host applications for servicing the memory access requests, and provide result/status information to the host applications in accordance with ~~the~~ a protocol of each of the number of host applications, wherein the memory interface comprises:

a host interface for interfacing with the number of host applications;

a memory interface for interfacing with the memory device;

a number of contexts operably coupled to the host interface for receiving memory access requests from the number of host applications providing result/status information to the number of host applications, wherein at least one context is provided for each host application; and

control logic operably coupled to obtain memory access requests from the number of contexts, interact with the memory device over the memory interface for servicing the memory access requests on behalf of the number of host applications, and provide the result/status information to the number of host applications via the number of contexts.

33. (cancelled) ~~The apparatus of claim 32, wherein the memory interface device comprises:~~

~~a host interface for interfacing with the number of host applications;~~

~~a memory interface for interfacing with the memory device;~~

~~a number of contexts operably coupled to the host interface for receiving memory access requests from the number of host applications providing result/status information to the number of host applications; and~~

~~control logic operably coupled to obtain memory access requests from the number of contexts, interact with the memory device over the memory interface for servicing the memory access requests on behalf of the number of host applications, and provide the result/status information to the number of host applications via the number of contexts.~~

34. (original) The apparatus of claim 33, wherein the number of host applications comprises a number of packet processing contexts of a packet processor contexts of a packet processor, and wherein the host interface conforms to a packet processor interface.

35. (original) The apparatus of claim 33, wherein the memory device comprises a content-addressable memory (CAM), and wherein the memory interface conforms to a CAM interface.

36. (original) The apparatus of claim 33, wherein the number of contexts comprises a number of context registers sets.

37. (original) The apparatus of claim 36, wherein each context register set corresponds to one and only one of the number of host applications.

38. (original) The apparatus of claim 33, wherein the control logic comprises:

monitoring logic;

scheduling logic;

memory interface logic; and

result/status logic, wherein:

the monitoring logic is operably coupled to monitor the number of contexts for detecting memory access requests and providing the memory access requests to the scheduling logic;

the scheduling logic is operably coupled to schedule memory access operations for the memory access requests;

the memory interface logic is operably coupled to generate memory interface signals for interfacing with the memory device over the memory interface; and

the result/status logic is operably coupled to provide result/status information to the number of host application(s).

39. (original) The apparatus of claim 38, wherein each context comprises a context register set, and wherein the monitoring logic is operably coupled to monitor a predetermined register in each context register set to detect a memory access request.

40. (original) The apparatus of claim 39, wherein the predetermined register comprises an instruction register.

41. (original) The apparatus of claim 38, wherein the memory interface supports pipelining of memory access operations, and wherein the scheduling logic is operably coupled to a pipeline a plurality of memory access requests over the memory interface.

42. (original) The apparatus of claim 41, wherein the scheduling logic is operably coupled to determine that a plurality of memory access requests conflict and execute at least one of the conflicting memory access requests as an atomic operation.

43. (original) The apparatus of claim 42, wherein the scheduling logic is operably coupled to clear the pipeline in order to execute the conflicting memory access request as an atomic operation.

44. (original) The apparatus of claim 38, wherein the result/status logic is operably coupled to correlate result/status information with its corresponding memory access request.

45. (original) The apparatus of claim 38, wherein the result/status logic is operably coupled to store the result/status information for each memory access request in a corresponding context.

46. (original) The apparatus of claim 45, wherein each context comprises a validity indicator, and wherein the result/status logic is operably coupled to set the validity indicator in each context when the corresponding memory access is complete and the result/status information is available.

47. (original) The apparatus of claim 32, wherein the memory interface device is a programmed programmable logic device.

48. (original) The apparatus of claim 32, wherein the memory interface device is an application specific integrated circuit.